Claims

We claim:

1. A timezone database structure, comprising:

at least one timezone identifier;

at least one local shift time associated with each said timezone identifier; and

an anchor shift time associated with each said local shift time.

2. The database structure of claim 1, wherein the local shift times associated with a given timezone identifier of the at least one timezone identifier are in sorted order.

3. The database structure of claim 1, wherein the local shift times associated with a given timezone identifier of the at least one timezone identifier are in unsorted order.

4. The database structure of claim 1, wherein a timezone associated with a timezone identifier of the at least one timezone identifier is non-observing.

5. The database structure of claim 1, wherein a timezone associated with a first timezone identifier of the at least one timezone identifier is observing.

6. The database structure of claim 5, wherein a timezone associated with a second timezone identifier of the at least one timezone identifier is non-observing.

7. The database structure of claim 1, wherein a first local shift time associated with a first timezone identifier of the at least one timezone identifier is relative to a reference date-time.

8. The database structure of claim 7, wherein the first local shift time is after the reference date-time.

9. The database structure of claim 7, wherein the first local shift time is before the reference date-time.

10. The database structure of claim 9, wherein a second local shift time associated with the first timezone identifier is relative to the reference date-time, and wherein the second local shift time is after the reference date-time.

11. The database structure of claim 1, wherein the at least one local shift time and the anchor shift time are expressed in a format selected from the group consisting of an integer format, a floating point format, an octal format, a hexadecimal format, a binary format, a character format, and combinations thereof.

12. A method of generating a timezone database structure, comprising the steps of:

generating at least one timezone identifier;

generating at least one local shift time associated with each said timezone identifier; and

generating an anchor shift time associated with each said local shift time.

13. The database structure of claim 12, wherein the steps of generating at least one local shift time and generating an anchor shift time include executing a Local/Anchor Shift Time (LAST) algorithm.

14. The database structure of claim 12, wherein the step of generating at least one local shift time includes causing the at least one local shift time to be in sorted order.

15. The database structure of claim 12, wherein the step of generating at least one local shift time includes causing the at least one local shift time to be in unsorted order.

16. The method of claim 12, wherein a timezone associated with a timezone identifier of the at least one timezone identifier is non-observing.

17. The method of claim 12, wherein a timezone associated with a first timezone identifier of the at least one timezone identifier is observing.

18. The method of claim 17, wherein a timezone associated with a second timezone identifier of the at least one timezone identifier is non-observing.

19. The method of claim 12, wherein a first local shift time associated with a first timezone identifier of the at least one timezone identifier is relative to a reference date-time.

20. The method of claim 19, wherein the first local shift time is after the reference date-time.

21. The method of claim 19, wherein the first local shift time is before the reference date-time.

22. The method of claim 21, wherein a second local shift time associated with the first timezone identifier is relative to the reference date-time, and wherein the second local shift time is after the reference date-time.

23. The method of claim 12, wherein the steps of generating at least one local shift time and generating an anchor shift time comprise expressing the at least one local shift time and the anchor shift time in a format selected from the group consisting of an integer format, a floating point format, an octal format, a hexadecimal format, a binary format, and a character format.

24. The method of claim 12, wherein the steps of generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include adding a new timezone identifier and associated local shift time data and associated anchor shift time data.

25. The method of claim 12, wherein the steps of generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include deleting a new timezone identifier and associated local shift time data and associated anchor shift time data.

26. The method of claim 12, wherein the step of generating at least one local shift time and generating an anchor shift time include adding a local shift time and an associated anchor shift time for an existing timezone identifier of the at least one timezone identifier.

27. The method of claim 12, wherein the steps of generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include:

replacing a timezone identifier that exists in the timezone database structure with a replacement timezone identifier,

replacing a local shift time that exists in the timezone database structure with a replacement local shift time,

replacing an anchor shift time that exists in the timezone database structure with a replacement anchor shift time, or

combinations thereof.

28. A method of using a timezone database structure to perform a timezone-related calculation, comprising the steps of:

providing the timezone database structure which includes at least one timezone identifier, at least one local shift time associated with each said timezone identifier, and an anchor shift time associated with each said local shift time; and

performing the timezone-related calculation utilizing the timezone database structure.

29. The method of claim 28, wherein the performing step includes utilizing a Local Time Conversion (LTC) algorithm to convert a local time of a first timezone to a corresponding local time of a second timezone, and wherein the first timezone and the second timezone each have a timezone identifier that is included in the timezone database structure.

30. The method of claim 29, wherein the first timezone is observing and the second timezone is observing.

31. The method of claim 29, wherein the first timezone is observing and the second timezone is non-observing, or wherein the first timezone is non-observing and the second timezone is observing.

32. The method of claim 29, wherein the first timezone is non-observing and the second timezone is non-observing.

33. The method of claim 28, wherein the performing step includes utilizing a Double Time Test (DTT) algorithm to determine whether a given local time in an observing timezone is in double time, and wherein the observing timezone has a timezone identifier that is included in the timezone database structure.

34. The method of claim 33, if the DTT algorithm determines that the given local time is in double time, further comprising converting the given local time to an associated anchor time, said converting including use of a double time offset.

35. The method of claim 28, wherein the performing step includes utilizing an Advanced Time Test (ATT) algorithm to determine whether a given date-time of an observing timezone is in Advanced Time, and wherein the observing timezone has a timezone identifier that is included in the timezone database structure.

36. The method of claim 28, wherein the performing step includes utilizing an Lost Time Test (LTT) algorithm to determine whether a given date-time of an observing timezone is in lost time, and wherein the observing timezone has a timezone identifier that is included in the timezone database structure.

37. A computer system for generating a timezone database structure, comprising:

a processor;

an input device coupled to the processor;

an output device coupled to the processor;

a first memory device coupled to the processor;

a second memory device coupled to the processor; and

a computer code stored in the second memory device and executed by the processor, said computer code comprising an algorithm for generating the timezone database structure, said algorithm comprising: generating at least one timezone identifier, generating at least one local shift time associated with each said timezone identifier, and generating an anchor shift time associated with each said local shift time.

38. The computer system of claim 37, wherein the algorithm includes a Local/Anchor Shift Time (LAST) algorithm for said generating the at least one local shift time and the associated anchor shift times.

39. The database structure of claim 37, wherein the algorithm causes the at least one local shift time to be in sorted order.

40. The database structure of claim 37, wherein the algorithm causes the at least one local shift time to be in unsorted order.

41. The computer system of claim 37, wherein a timezone associated with a timezone identifier of the at least one timezone identifier is non-observing.

42. The computer system of claim 37, wherein a timezone associated with a first timezone identifier of the at least one timezone identifier is observing.

43. The computer system of claim 42, wherein a timezone associated with a second timezone identifier of the at least one timezone identifier is non-observing.

44. The computer system of claim 37, wherein the algorithm generates a first local shift time associated with a first timezone identifier of the at least one timezone identifier, and wherein the first local shift time is relative to a reference date-time.

45. The computer system of claim 44, wherein the first local shift time is after the reference date-time.

46. The computer system of claim 44, wherein the first local shift time is before the reference date-time.

47. The computer system of claim 46, wherein the algorithm generates a second local shift time associated with the first timezone identifier, and wherein the second local shift time is relative to and after the reference date-time.

48. The computer system of claim 37, wherein the algorithm expresses the at least one local shift time and the anchor shift time in a format selected from the group consisting of an integer format, a floating point format, an octal format, a hexadecimal format, a binary format, and a character format.

49. The computer system of claim 37, wherein generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include adding a new timezone identifier and associated local shift time data and associated anchor shift time data.

50. The computer system of claim 37, wherein generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include deleting a new timezone identifier and associated local shift time data and associated anchor shift time data.

51. The computer system of claim 37, wherein generating at least one local shift time and generating an anchor shift time include adding a local shift time and an associated anchor shift time for an existing timezone identifier of the at least one timezone identifier.

52. The computer system of claim 37, wherein generating at least one timezone identifier, generating at least one local shift time, and generating an anchor shift time include:

replacing a timezone identifier that exists in the timezone database structure with a replacement timezone identifier,

replacing a local shift time that exists in the timezone database structure with a

replacement local shift time,

replacing an anchor shift time that exists in the timezone database structure with a replacement anchor shift time, or

combinations thereof.

53. A computer system for using a timezone database structure to perform a timezone-related calculation, comprising:

a processor;

an input device coupled to the processor;

an output device coupled to the processor;

a first memory device coupled to the processor;

a second memory device coupled to the processor;

the timezone database structure coupled to the processor, said timezone database structure comprising: generating at least one timezone identifier, generating at least one local shift time associated with each said timezone identifier, and generating an anchor shift time associated with each said local shift time; and

a computer code stored in the first or second memory device and executed by the processor, said computer code comprising at least one algorithm, said algorithm comprising a capability for performing the timezone-related calculation utilizing the timezone database structure.

54. The computer system of claim 53, wherein the at least one algorithm includes a Local Time Conversion (LTC) algorithm to convert a local time of a first timezone to a corresponding local time of a second timezone, and wherein the first timezone and the second timezone each have a timezone identifier that is included in the timezone database structure.

55. The computer system of claim 54, wherein the first timezone is observing and the second timezone is observing.

56. The computer system of claim 54, wherein the first timezone is observing and the second timezone is non-observing, or wherein the first timezone is non-observing and the second timezone is observing.

57. The computer system of claim 54, wherein the first timezone is non-observing and the second timezone is non-observing.

58. The computer system of claim 53, wherein the at least one algorithm includes a Double Time Test (DTT) algorithm to determine whether a given local time in an observing timezone is in double time, and wherein the observing timezone has a timezone identifier that is included in the timezone database structure.

59. The computer system of claim 58, if the DTT algorithm determines that the given local time is in double time, further comprising converting the given local time to an associated anchor time, said converting including use of a double time offset.

60. The computer system of claim 53, wherein the at least one algorithm includes an Advanced Time Test (ATT) algorithm to determine whether a given date-time of an observing timezone is in Advanced Time, and wherein the observing timezone has a timezone identifier that is included

in the timezone database structure.

61. The computer system of claim 53, wherein the at least one algorithm includes an Lost Time Test (LTT) algorithm to determine whether a given date-time of an observing timezone is in lost time, and wherein the observing timezone has a timezone identifier that is included in the timezone database structure.

62. A computer program product, comprising a computer usable medium having a computer readable program code embodied therein, wherein the computer readable program code comprises an algorithm which generates a timezone database structure, said algorithm comprising: generating at least one timezone identifier, generating at least one local shift time associated with each said timezone identifier, and generating an anchor shift time associated with each said local shift time.

63. The computer program product of claim 62, wherein the algorithm includes a Local/Anchor Shift Time (LAST) algorithm for said generating the at least one local shift time and the associated anchor shift times.

64. A computer program product, comprising a computer usable medium having a computer readable program code embodied therein, wherein the computer readable program code comprises an algorithm which uses a timezone database structure to perform a timezone-related calculation.

65. The computer program product of claim 64, wherein the algorithm is selected from the group consisting of a Local Time Conversion (LTC) algorithm, a Double Time Test (DTT) algorithm, an Advanced Time Test (ATT) algorithm, an Lost Time Test (LTT) algorithm, and combinations thereof.